

## Justificación del documento

Tras la reunión de la semana pasada me di cuenta que no sabemos muy bien qué es lo estamos implementando y esto nos lleva a pegar palos de ciego y hacer cosas que no deberíamos de hacer.

Quizás el problema es que nos hemos puesto a implementar un FK genérico (aunque luego particularizado para unas determinadas dimensiones)

Para poder centrar el problema hemos pensado que lo mejor es fijar un determinado modelo de proceso y de medición, a partir de los cuales extraeremos las ecuaciones del filtro kalman que finalmente implementaremos. Para que nos sirva todo lo que lleváis implementado hasta el momento he buscado un artículo que en el que la matriz A tenga unas dimensiones de 3x3

El artículo que he seleccionado es el que aparece en la cabecera de este documento, el de Palermo del año 2005. La nomenclatura que utiliza es diferente a la que estamos siguiendo nosotros (Bishop 2001), por lo que a lo largo del documento pondré la información necesaria para no liarnos.

Además he pensando ir añadiendo al documento temas relacionados con la implementación del filtro y con el uso de la herramienta. De esta manera tendremos todos un guión que seguir y que estructure y homogenice el trabajo

# El filtro kalman

## Modelos de proceso y de medición

Fijar los modelos de proceso y de medición es de vital importancia para la extracción posterior de las ecuaciones del filtro. Estos modelos son los que determinan si el filtro será discreto o extendido y las dimensiones de las matrices. Palermo utiliza los siguientes modelos. Para el proceso







Donde :

* g🡪 la glucosa intersticial
* d🡪la velocidad de variación de la glucosa intersticial. Por lo tanto d es la derivada primera de g
* f🡪la velocidad de variación de d, por lo tanto f es la derivada primera de d y la derivada segunda de g
* w es el ruido del proceso que finalmente representaremos por la covarianza del ruido del proceso Q

el modelo de medición que utiliza es



Donde :

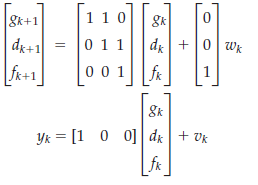
y🡪 es el valor medido (que en Bishop se denomina z)

g🡪 la glucosa intersticial

v🡪 el ruido de la medición que finalmente lo expresaremos como la covarianza del ruido de la medición R

**La primera información que extraemos de estos modelos es que ambos son lineales por lo tanto el filtro que tenemos que implementar es el DISCRETO NO EL EXTENDIDO**

A continuación representamos estos modelos en forma matricial:



Lo primero que debemos observar es que en estas expresiones NO aparece el término de Bishop Buk.. Esto se debe a que el modelo de Bishop es un modelo general y supone que existe una entrada de control (justamente uk), pero en los filtros aplicados al control de glucosa no se utiliza entrada de control por lo que el término Buk de la expresión desaparece.(esto ya implica una modificación en las expresiones que estamos utilizando)

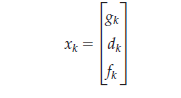
Si queremos poner estas ecuaciones en forma compacta vemos que

Xk=Axk-1+wk

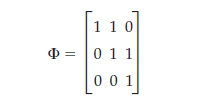
zk=Hxk+vk (Palerm la llama y)

que concuerdan con las expresiones 4.1 y 4.2 de Bishop 2001, salvo por el término correspondiente a la entrada de control que hemos eliminado

Por lo tanto tenemos, que x es un vector 3x1 que contiene los 3 estados del proceso



A es una matriz 3x3 ( Palermo la llama ɸ)



H es una matriz(1x3) (Palermo la llama C)



Sabiendo que las dimensiones generales de un modelo son

X-->nx1

A-->nxn

B-->nxl

H-->mxn

Z-->mx1

KI-->nxm

Q-->nxn

P-->nxn

R-->mxm

Y fijándonos en la dimensiones de A,C z y x tenemos que

X-->3x1

A-->3x3

B-->vale 0, no aparece el término

H-->1x3

Z-->1x1

K-->3x1🡪 Palermo la llama L

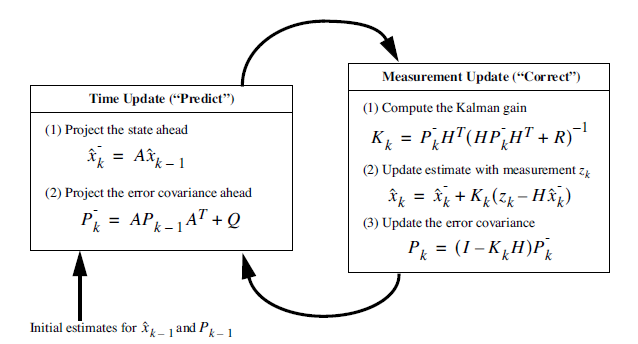
Q-->3x3

P-->3x3

R-->1x1🡪de aquí extraemos que no hace falta hallar la matriz invertida

## Expresiones del filtro kalman discreto

Con todo lo anterior las ecuaciones del filtro , según la nomenclatura de Bishop son



## Equivalencias de nomenclaturas:

|  |  |  |
| --- | --- | --- |
|  | Palermo | Bishop |
| estados | x | x |
| medición | y | z |
| matriz de relación entre xk y xk-1 | ɸ | A |
| matriz de relación entre el estado y la medida | C | H |
| ganancia del filtro | L | K |
| covarianza del ruido del proceso | Q | Q |
| covarianza del ruido de la medición | R | R |
| covarianza del error asociada a la estimación | P | P |

Como podéis observar utilizando como filtro a implementar el propuesto por Palermo, todo queda muy parecido a lo que estábamos haciendo pero ligeramente más sencillo y debidamente justificado

# implementación

## Comunicación entre los módulos del diseño

En un principio se decidió que la comunicación entre módulos se realiza entre fifos.

En este apartado vamos a fijar :

1. Como es el protocolo de comunicación entre fifos
2. Tamaño de la información que se intercambia entre los diferentes módulos

### Protocolo de comunicación:

En construcción

### Tamaño de la información que se intercambia.

Me da la sensación que estamos presuponiendo que el tamaño de la información que se intercambia entre los diferentes módulos es siempre la misma: matrices de 3x3 por lo que implementamos fifos de tamaño 9. Pero la información que se intercambia es no siempre es de 3x3 lo que afecta al tamaño de las fifos y/o a los protocolos de comunicación:

A continuación añado tres secciones una por módulo uno para cada módulo, indicando cual son las entradas/salidas, su tamaño , origen/destino, nombre dado en el diseño

#### Modulo estim:

Este modulo es el que hace las predicciones:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| entrada | tamaño | origen | nombre\_diseño | observaciones |
| A | 3x3 | ¿? | c\_rom\_a | es una constante, no figura como entrada en el módulo |
| x | 3x1 | correct | x\_post\_data |  |
| P | 3x3 | correct | p\_post\_data |  |
| Q | 3x3 | ¿? | q | ¿de dónde viene?¿es una constante? |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| salidas | tamaño | destino | nombre\_diseño | observaciones |
| x­- | 3x1 | correct | x\_priori\_data |  |
| P- | 3x3 | gain/correct | p\_priori\_data |  |

### Modulo gain

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| entrada | tamaño | origen | nombre\_diseño | observaciones |
| P- | 3x3 | predict | p\_priori\_data |  |
| H | 1x3 | ¿? | c\_rom\_H | definido como constante, no figura como entrada en el módulo |
| R | 1x1 | ¿? | r | no está definido como constante, no se donde la incluiremos |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| salidas | tamaño | destino | nombre\_diseño | observaciones |
| K | 3x1 | correct | gain\_data |  |

### Modulo correct

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| entrada | tamaño | origen | nombre\_diseño | observaciones |
| x- | 3x1 | estim | x\_post\_data |  |
| K | 3x1 | gain | gain\_data |  |
| z | 1x1 | lectura | input\_data | este dato viene de una medida exterior |
| H | 1x3 | ¿? | c\_rom\_H | definido como constante, no figura como entrada en el módulo |
| P- | 3x3 | estim | ¿? | no la veo definida en el módulo |
| I | 3x3 | ¿? | ¿? | no la veo definida |
|  |  |  |  |  |

## tarea: Cambio de nombre de los módulos

Los nombres de los módulos que habéis elegido son un poco confusos por ejemplo el modulo update crea confusión porque según la terminología de bishop hay dos updates:

* Time update
* Measurment update

En el caso de estimation no sigue ninguna de las terminologías de bishop

Tenéis cambiar los nombres a los siguientes:

|  |  |
| --- | --- |
| ANTIGUO | NUEVO |
| estimation | prediction |
| update | correction |
| gain | gain |

# herramientas

## Simulación de componentes

Hay que distinguir dos casos:

* El componente solo necesita las señales de ck, rst y/o start
* El componente necesita las señales clk, rst , start y datos de entrada

En el primer caso basta con crear un nuevo componente de tipo diagrama del bloques con el prefijo tb\_ al que se le añaden

* el bloque a simular
* los bloques creados por óscar que generan las señales de reloj , reset y start ((por ejemplo mirar el tb\_fifo\_read)

En el segundo caso seleccionas el componente a simular (que llamaré verificar) y seleccionas

file🡪new🡪test bench

en la ventana de dialogo que aparece seleccionas la opción graphical

Aparece un nuevo componente con el sufijo\_tb (verificar\_tb) que es un diagrama de bloques con dos componentes:

* verificar
* verificar\_tester

el bloque verificar\_tester tiene como salidas las entradas al bloque verificar y como entradas las salidas del bloque verificar

Pinchar dos veces en el veirficar\_ tester y en el cuadro de dialogo selecciona vhdl

Una vez hecho esto ir al directorio TFG2015\src\tb\_design\tb\_common\hdl y abrir el archivo tb\_generic.vhd

Este archivo contiene un ejemplo de cómo leer/ escribir datos de/a un archivo .txt. Dentro del archivo viene instrucciones de cómo se debe modificar el archivo para que se adapte .

Copiar el archivo tb\_generic en verificar\_tester y modificar todo lo que se indica en el archivo

Añadir los bloques de start, reset y clk